Enhanced IoT security through orchestrated policy enforcement gateways

Hamza Attak, Ludovic Jacquin, Adrian L. Shaw, Marco Casassa-Mont and Yolanta Beresna

Hewlett Packard Labs, Security and Manageability lab, Bristol (UK).

{hamza.attak, ludo, als, marco_casassa-mont, yolanta.beres}@hpe.com

Abstract. The increasing number of IoT devices raises concerns about the amount of data they generate and – more importantly – their content, having security and privacy implications. The Things are mostly constrained by typical embedded design limitations from non-extensible functionalities to poor or non-existent configuration; adding security features to these devices is therefore impractical. This paper presents a network security infrastructure suitable for IoT devices, which aims at offloading the security from the devices to the nearest network edge they are connected to. First, the SECURED architecture for the network edge device (NED) is detailed: its components, security policy refinement and translation, and the way it addresses mobility of the things. Then, the SHIELD architecture proposes to extend and strengthen the security of the IoT devices by leveraging the dynamic deployment of security controls with analytics, which permits orchestrated security at the entire infrastructure level – allowing a new threat detection paradigm.

Keywords: SDNFV infrastructure, network security, trusted computing, big data
Category: specialised.

1 Introduction

The rate of new devices and platforms connecting to the internet is causing concern throughout the security industry. Firstly, the number of different platforms to secure is akin to the mobile-device-management problems already faced today, only the quantity is far larger and capability gaps between devices is even greater. Secondly, some IoT devices need to operate within such tight operational constraints (such as power or form-factor restrictions) that even if manufacturers wanted to add sufficient security features – such as data or traffic encryption, access control, etc. –, they could not. Finally, the heterogeneity of the IoT landscape, and the potential size of its deployment, requires an automated management of its security.

This paper presents two solutions to these challenges by leveraging the Network Function Virtualisation (NFV) – and the underlying Software-Defined Network (SDN)

– paradigms. Specifically, section 2 presents the two main results of the EU SECURED project [1], namely, (i) the security-enhanced NFV infrastructure which allows to offload some IoT security into the edge of the network they are connected to, (ii) the solutions for an easy to use security: high level security policies – and the translation services to transform them into enforceable low level configuration –, but also (iii) a mobility-aware solution – whereby the infrastructure can move the security controls to follow the IoT devices. Section 3 focuses on the security improvement allowed by flexible deployment of security controls in the network, coupled with analytics which combine multiple security controls dynamically to secure the end devices; these improvements will be demonstrated in the EU SHIELD project. Section 4 discusses the results obtained during the research work. Section 5 concludes this paper, highlighting the major contribution and next steps of these two projects.

2 SECURED: security at the network edge

Through the presentation of the SECURED project, this section proposes solutions to enforce security compatible with IoT devices, which avoids altering the actual devices, by offloading the security at the network edge. In this context, SECURED focuses on three main goals: (i) ensure the security of IoT devices; (ii) provide an easy-to-use and configurable way to specify the security policy for a user or an IoT device; (iii) and define a mobility paradigm for geographically moving subjects (such as wearables or drones for instance).

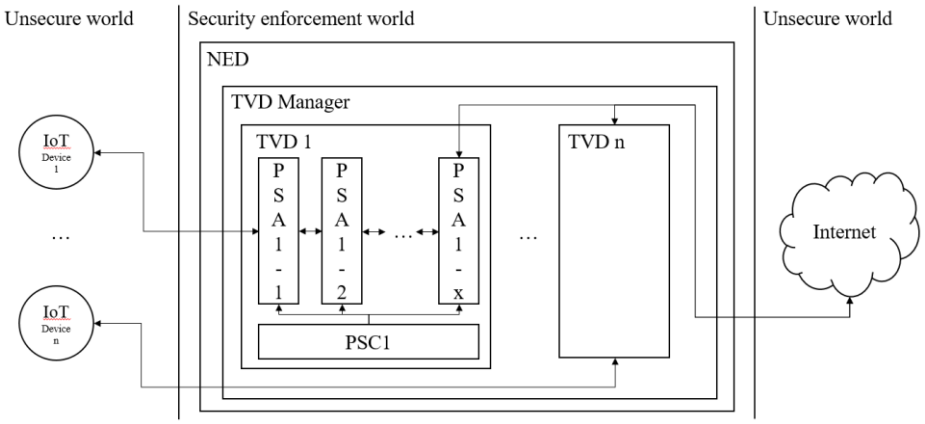

Fig. 1. Overview of the NED and its component

2.1 The SECURED architecture and design

SECURED is a complete solution for securing existing IoT devices and networks which strongly emphasises on usability: it enforces the security at the nearest network

edge instead of on the device itself. Standing between the devices and the outer – unsecured – network, it can therefore not be bypassed. Moreover, the security being offloaded from the Thing, it does not require any modification of the client devices.

It is architectured around three main components in order to answer the security challenges: the Network Edge Device (NED), the Trusted Virtual Domain (TVD) and the Personal Security Application (PSA). An overview of the architecture and the different components' interactions can be seen on figure 1; each of these components is detailed in this section, starting with the PSA concept up to their organization.

In SECURED, security is accomplished in a software-driven fashion, where instances of security applications are dynamically deployed to perform their specific function. These functions are represented in the architecture by the PSAs: a PSA being a self-contained application that performs one or more security functions.

A PSA enforces a set of security functions, but a device might need more than one PSA to achieve its security requirements. The TVD concept embraces the set of chained PSAs needed by a tenant (similarly to a network service chain in ETSI NFV [2]) and combine it with a Personal Security Controller (PSC). TVDs are tenant specific and provide isolation of the security for each IoT device. The order and configuration of the PSAs inside a TVD are critical to enforce the correct tenant security: it is defined through a policy model, which is described in section 2.3. The PSC represents the controller of the tenant PSAs; it allows to continuously dialogue and monitor them.

Finally, the NED brings together all the previous components into an instantiation of the SECURED concept. Physically, it is the device – or set of devices in the case of a distributed deployment – that gates the Things from the unsecure outer network. It hosts the TVD of every IoT device connected to it.

The TVDs are managed by a TVD manager whose role is to create, destroy the TVDs and also to monitor – through to the PSC – that all the TVDs are working correctly.

Following the SDN model, a control plane and a data plane are setup in order to isolate the different communication flows occurring inside the NED. The data plane is represented by the PSAs and their chaining with the tenant. The control plane is present at two locations in SECURED: (i) between the TVD manager and each TVD (actually through its PSC), and (ii) between the PSCs and their associated PSAs.

The NED being the point where the security functions are executed, it has to be trusted by the tenants. SECURED leverages trusted computing techniques such as remote attestation [3] with the use of discrete Trusted Platform Module (TPM) to ensure of the identity of the NED and the state of its software stack – validated against known-good states –; SECURED can also verify the SDN rules applied by the network switches [4] in the case of the user's traffic being redirected (e.g. when the current NED cannot enforce every PSAs of the user).

2.2 Implementation

In practice, PSAs can be implemented as virtual machines (VM) or containers. There are two technical requirements for a PSA: first, a PSA needs to be isolated from the other components (i.e. PSAs, PSCs, TVDs, TVD manager) running on the same NED's

host; second, a PSA need at least three network interfaces: two defined as chain-head and chain-tail, in order to plug the PSA between the device to secure and the outside unsecure network; another one is reserved in order to control the PSA from the TVD – through the PSC –, representing the control plane.

Similar to PSAs in practice (VM or container), the role of a PSC is to control and monitor the PSAs inside the same TVD. It is also the bridge between the TVD manager, running on the host system of the NED, and the PSAs. Its main function is to start, stop, configure or get the status of PSAs. It is the abstraction layer for interacting with the TVD (either by the tenant or by the TVD manager). Similar to a PSA, the PSC must respect storage and execution isolation on the system it is running on. It also needs two network interfaces: one to control the different PSAs and one to receive higher level requests from the TVD manager running on the host system. The PSC is only present on the control plane.

The TVD being the association of a set of PSAs and one PSC, it is defined mostly as a conceptual abstraction. Nonetheless, its practical parts appear at TVD runtime. As a matter of fact, when the TVD manager instantiates a TVD, it has to create all the networking links between the PSAs – representing the data plane service chain – and attach itself to the PSC in order to manage it.

The TVD manager represents the isolation layer between the different TVDs - and associated components; it also includes the management components for creating and managing the TVDs. In practice, it can be implemented through a virtualisation hypervisor or a container engine.

The SECURED architecture ensures isolation between TVDs in order to respect privacy between users. In practice, it is mainly done by the fact that PSAs isolation is already guaranteed by the hypervisor or container engine and the PSA chain is ensured to be linked without leaks with a well-understood and trusted network configuration. For more advanced TVDs topologies, human verification does not scale. There is a need for automated verification in more dynamic and complex virtual network configurations to ensure that isolation is applied properly.

Open vSwitch [5] and OpenFlow [6] technologies are used for their flexibility in order to dynamically chain the PSAs together and therefore create a TVD between a given IoT device connected to a specific port and the outer unsecured requested resource.

2.3 Policy Model

SECURED can enforce security at the edge for any device or user with multiple devices and applications. In the context of this paper, the security for an IoT device can be represented as nothing else but a TVD instantiated for a specific user having a single device. As a user-centric concept, it is therefore very important to provide good interfaces to interact with the SECURED components. These interfaces and their organization are the subject of this section.

As security functions, PSAs need to be configured to work correctly. One of the problem between non-technical users and software in general is that it is a tedious task to configure software. This statement is even truer with security software.

Through SECURED, a high-level way to define security policies is proposed. Simply called the High-level Security Policy Language (HSPL), it is aimed to be very close to the natural language.

The challenge here is to provide HSPL for the user and at the same time achieve a coherent security chain in the TVD with this high level configuration.

For the sake of flexibility and compatibility, SECURED does not intend to modify the security applications running on the PSAs to be compatible with HSPL (even though the PSAs still need to be configured with their specific syntax and languages). Instead, one more level of abstraction is introduced in order to bridge HSPL with the low level PSA-specific configuration. This is the Medium-level Security Policy Language (MSPL).

Its goal is to extract from the HSPL, the different security capabilities required by the policy. It also set up an application graph which is a proposal of a service chain with different available PSAs to realise the security policy defined by the user. Anomalies can occur at this stage, SECURED warns the user if one happens (unavailable PSAs for example).

The interactions between the different abstraction levels can be seen on figure 2.

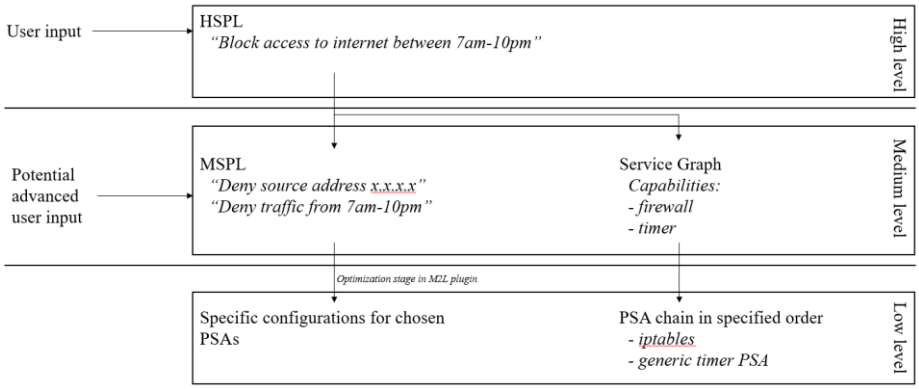

Fig. 2. Workflow example for a given policy

In order to propose PSAs coherent with the defined high level policy, a set of security capabilities is defined beforehand. The HSPL implements these capabilities and each PSA exposes its set. Depending on the capability, a PSA will be place before another on a specified service chain. For example, it makes no sense to put a traffic encryption PSA at the beginning or in the middle of a chain.

SECURED also tries to address the scaling problem. With this idea in mind, SECURED authorizes the network administrators or advanced users to modify their MSPL configuration and specify the specific PSAs they want to use or their orders. Of course, in this situation, the coherency and correctness of the PSAs security chain is not guaranteed by the architecture. Nonetheless, SECURED could still detect certain anomalies on the edited service chain and therefore warn the user (encryption PSA not at the end of a chain for example, or PSAs with redundant capabilities in the chain).

Different constraints can appear when a user is setting up his policy. These constraints are the results of higher priority policies that can be set up by a higher authority entity (for example, parents for children, company for employees…). Before translating the MSPL to low level configurations for the PSAs, a policy reconciliation step is to be conducted in order to verify the compliance of the user-defined policy with the potential higher priority policies (the reconciliation process detailed in [7]). Before applying the policy, its reconciled version is displayed to the user on the SECURED configuration dashboard.

On one hand, the application graph is specifying which PSAs to instantiate and their order, on the other hand, their configuration is still to be translated from the generated MSPL. In order to facilitate the translation, a configuration template is defined for each PSA. This permits the translation process to only fill in the variable parts of the configuration. The PSA developer is responsible for providing this template and the translation plugin between the standardized MSPL and the low-level PSA-specific configuration. The same way SECURED is device-agnostic, this also makes it application-agnostic.

2.4 Mobility scenario

Given the dynamic nature of some IoT deployment, SECURED proposes a solution to maintain a user security context throughout the different NED a device can connect to. When a secure channel to a new NED is established, SECURED detects the user profile and immediately starts to migrate its TVD from the last NED to the new one, which allows a seamless handover from one NED to another.

Let an IoT device wirelessly connected to a SECURED node A with a certain signal strength. Let it be moving away from A and therefore detects a decrease in node A's signal. From this situation, three cases are defined in the mobility context:

- The device meets another SECURED access point, in order to maximize performance, the SECURED architecture will start to transfer the tenant's TVD configuration to the second access point (see figure 3). It permits to have the newly instantiated TVD fully ready for when the new device will connect in order to cut down initialization and starting delays. This can be triggered by comparing signal strength and starting the transfer when the first access point's signal is under a defined threshold and the second access point's signal is strong enough.
- The device does not meet another SECURED access point, the device will simply lose connection and the security features added by SECURED. The tenant's TVD will be destroyed as soon as the connection with the SECURED node is terminated.
- The device meets another SECURED but never connects to the new access point. In that case, the device is still connected to the first access point but its TVD has started to be migrated to the next access point. In order not to keep its unused TVD on the second access point, a timeout is defined so that if the device does not transfer to the new TVD, it is automatically destroyed to avoid wasting resources on the second access point.

This description can be generalized to scale to more than two access points. As a matter of fact, the TVD configuration can be downloaded to all the nearby SECURED nodes or a more sensible solution would be to only transfer the configuration to the second strongest access point in terms of signal.

Also it has to be noted that the mobility scenario implies two major points: (i) the IoT device must be updated with the access point decision algorithm, it is therefore intended to more capable type of devices; (ii) there must a repository of tenant's configurations and their PSAs available to the SECURED nodes, it is out of scope of this paper but it is part of the SECURED project and is described in [1].

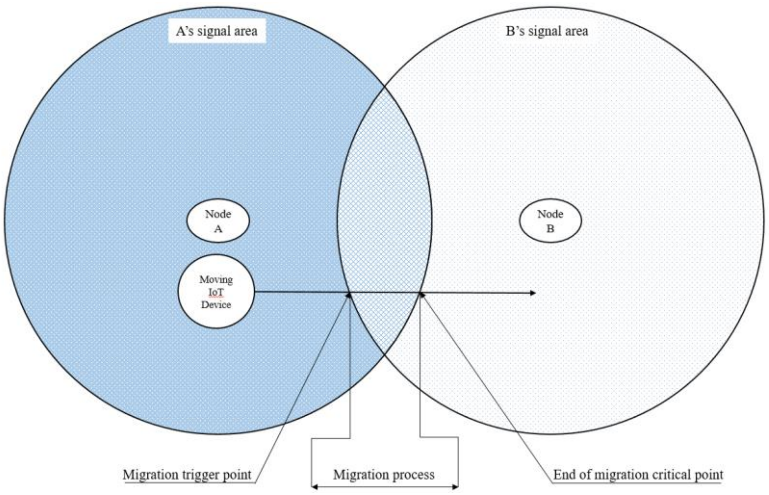

Fig. 3. Migration stages in the mobility scenario

3 SHIELD: leveraging big-data analytic for flexible security

SHIELD introduces big data analytics and remediation capabilities, on top of the SECURED infrastructure, to detect both known and unknown security attacks and remediate them. The core goal of SHIELD is to leverage a network-wide security view so that it can address distributed attacks such as Advanced Persistent Threats (APT).

In SHIELD, IoT devices consist of a very high number of distributed networked devices that connect to the network edge devices (e.g. in campus size networks). As previously mentioned, these IoT devices might lack basic security capabilities and rely on the underlying networking capabilities to share with other devices the data they locally collect (and potentially process).

In SHIELD, the edge networking devices are instrumented to further collect information about IoT devices, in particular about their networking and behaviour. To achieve this, they run virtual Network Security functions (vNSFs – conceptually similar to SECURED PSAs) under the control of a user/administrator and are orchestrated by SHIELD Management & Orchestration modules. They collect overall networking events (e.g. netflow, DNS, etc.), including the one generated by IoT devices, locally

process it and share it with centralized big data analytics solutions. They also play a key role as threat remediation end-points whereby actions to mitigate security risks and attacks are executed within these devices, with impact on locally networked systems.

The current high-level SHIELD architecture is shown in the following figure:

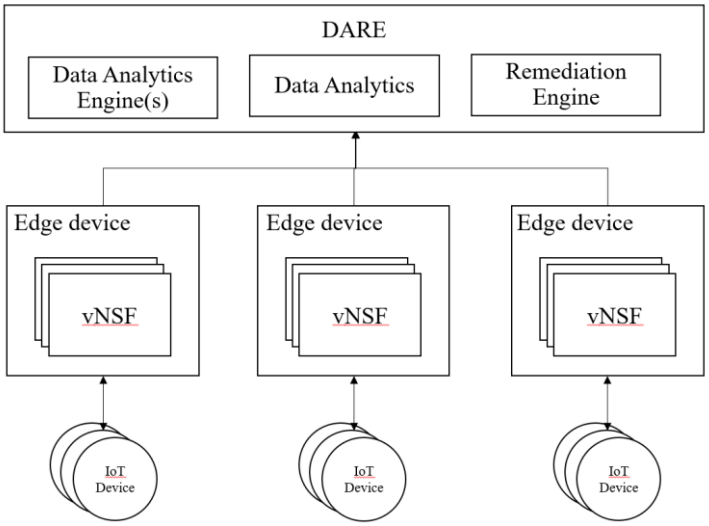

Fig. 4. High-level SHIELD architecture

The IT Infrastructure, inclusive of distributed edge devices, hosts and networks, represents the controlled and managed environment in SHIELD. Work done in SECURED, with regards to infrastructure attestation and verification, is further applied to SHIELD to continuously check for the integrity of edge devices, hosts, software and vNSFs.

On top of the IT Infrastructure, SHIELD runs a standard NFV Infrastructure, compatible with the one described in ETSI NFV Reference framework and architecture [3].

Specifically, SHIELD provides a set of vNSFs (as instances of Virtual Network Functions – VNFs) to support the collection of security events at the network level and security remediation activities (vNSFs Layer). Finally, a centralized, big data analytics solution is provided (via a set of analytics engines) to process collected data, monitor systems detect known/unknown threats and instruct remediation engines to remediate/recover attacks.

The following figure provide additional details about SHIELD components:

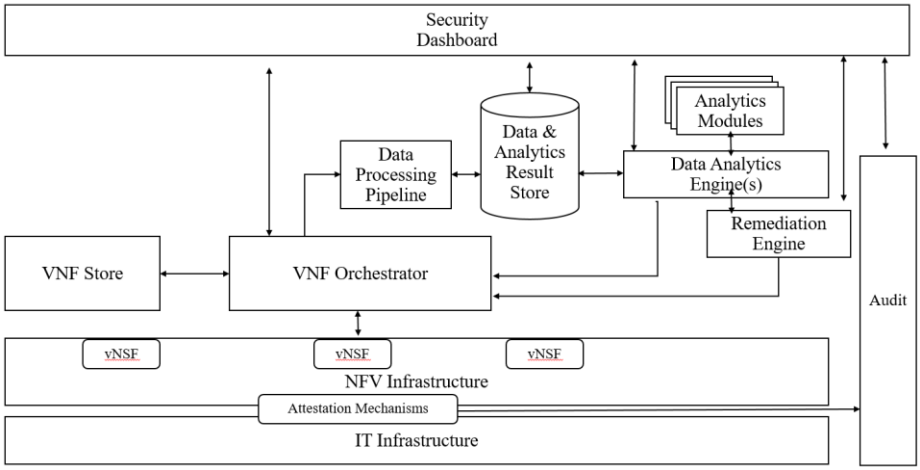

Fig. 5. SHIELD architecture components

The big data analytics solution consists of: network data/event collection points; data warehouse; analytics engines; a library of security analytics and UI.

It interacts with the underlying infrastructure via a VNF Orchestrator which also coordinates a VNF Store, storing available vNSFs functions.

Data collection points are implemented by a set of vNSFs which collect specific types of networking traffic (from IoT devices) via packet capturing techniques, including DNS events, netflow, DHCP, etc. These events are cleaned, enriched with additional metadata (e.g. geo-location of IP addresses, flagging suspicious IP addresses and domains based on threat intelligence, etc.) and stored in high-performance data repositories (inclusive of SQL and noSQL ones). Open sources event brokering and processing frameworks, including Apache Kafka [8] and Apache Storm [9] are used in the data processing pipeline.

A set of Analytics Engines processes the collected data both in near-time and on a historical basis (e.g. spanning from data collected in the last few hours back to weeks/months). They include rule engines and Apache Spark-based analytics engine.

These engine supports a wide range of threat detection mechanisms, driven by pattern-based analytics and machine learning analytics. The former checks for well-known attack patterns and mechanisms, driven by deep security knowledge and expertise in the field; the latter uses machine learning techniques to identify abnormal entity and user behaviours. Both types of analytics detect new security threats that happens at different stages of complex, advanced attacks, during potentially long time periods (weeks/months), including: initial compromise of a system/device; command & control by an external attacker site; lateral movements within an organization network; exploitation/damage.

Alerts are triggered from detected threats, further aggregated (e.g. across various engines) to determine their level of evidence/relevance and visualized to security analysts via a Security Dashboard UI.

A Remediation Engine is fed with top priority alerts and contextual information to determine a plan for mitigating existing threats and risks. This engine uses the alert details and context (e.g. infected device, network location, business priority, etc.), coupled with existing playbooks (workflows of steps and actions) to determine which remediation actions need to be carried out. Examples of actions might include: getting security personnel authorization to automatically remediate the threat; contact device owners; intervene at the networking level e.g. by blocking networking flows, redirecting them or enabling further logging activities at the networking level. Network-level remediation level automatically happens when the Remediation Engine contacts the NFV Management & Orchestration modules and activates relevant vNSFs to carry out the desired networking activities.

Finally, and Audit Service is used in SHIELD to log various collection, detection and remediation steps to provide an audit trail for future forensic analysis and incident management purposes.

The end-to-end security approach provided in SHIELD ensures that known/unknown security threats within an organisation are quickly detected by using trustworthy data, collected from a programmable NFV infrastructure and automatically remediated (or risks mitigated) by using the same NFV infrastructure and programmable vNSFs functions. The key value proposition is drastically reducing the time needed to remediate an attack, from its initial detection, hence reducing risk exposure and damages within an organisation.

4 Evaluation

In order to demonstrate the SECURED architecture, a NED prototype is implemented [10] towards an Intel Core i7-4770@3.40GHz (4+4 cores), 32GB of RAM with Fedora 20 integrating a 3.18.7-100.fc20.x86_64 kernel. The prototype also runs Open vSwitch in order to dynamically configure the network architecture (principally the TVD) between the NED internal components. PSAs are instantiated KVM-based VM or Docker images.

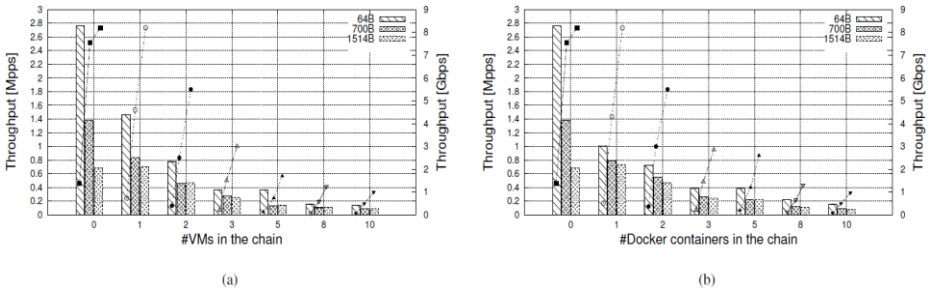

Fig. 6. Throughput of a single chain (of growing length) with VNFs deployed in: (a) KVM-based virtual machines; (b) Docker containers.

Figure 6 shows the throughput of a service chain depending on the PSA nature, the packet size and the chain length. This benchmark shows that virtualization is slightly faster than the container-based solution. It also shows that the throughput is inversely proportional to the number of running PSAs. This problem can be slightly overcome by grouping the device type into one chain, one TVD, and avoid of instantiating one chain per device. This also eases the administration of IoT devices as they are mainly data-centric. An example would be to group all the temperature sensors together so that they share they are secure the same way. This extension could easily ease the computing load on the SECURED node and need no adaptation whatsoever.

On the contrary, latency is reduced when using Docker. This is explained by the overhead implied by the virtualization and the fact that Docker has access the actual kernel driver functions whereas the packets have to cross two kernel stack in virtualization.

A second NED prototype also exists, it is built around an Intel Core-i5 3427U processor, 16GB of RAM and M.2 128GB flash storage. It runs a very thin and customized version of Debian along with Xen hypervisor. It has a very small form factor (seen on figure 7) and shows the NED as a home router appliance.

Fig. 7. 4"x4" NED prototype (~10.1x10.1 cm²)

Concerning the mobility scenario, Montero et Serral-Gracià [11] shows that the connection handover and the migration of a chain of two 512 MB PSAs variates from 7 to 10 seconds, and an average of 7 seconds without the migration. In fact, the connection handover delay is inherent and cannot be avoided, but the PSAs can either start to be transferred during the connection handover or their image not transferred at all, as only the configuration could be downloaded from the first access point to the second.

At the time of writing this paper, the SHIELD project has just begun and can therefore not provide results or a prototype. Nevertheless, the recent DDoS attack using a large number of IoT devices is the perfect example of what SHIELD on top of SECURED is protecting against. The NEDs and the DARE infrastructure enables SHIELD to aggregate the IoT devices network behaviour in a single decision component. This allows to make network wide decisions on different kind of attacks; for example in the recent DDOS IoT-based attack, SHIELD could have leverage a simple

analytics looking for distributed sources trying to connect to the same destination. Based on a specified threshold, some malicious connections could have been terminated at the NED level, protecting the core network and the destination from the flooding.

Similarly, the vNSFs are able to provide protection to those IoT devices connected to a managed network edge and could have prevented their tampering through the network in the first place.

5 Conclusion

This paper presents a security architecture for IoT devices, whereby security controls are hosted as close as possible to the user's device: at the network edge. SECURED results show how to bridge the trust gap of offloading security to the network, define a high-level common language to help setting up policies and translate them into enforceable configurations; and address the mobility challenge inherent to the IoT world.

Building on the SECURED architecture for running security controls in the network, the SHIELD vision is presented: the agile deployment of security functions, allowed by the use of a NFV-based infrastructure, enables the development of new security analytics that can orchestrate the security applications to either capture more (or less) security events, or address the attacks by deploying the appropriate remediation functions.

These new security paradigms do not rely on the end devices to embed heavy security mechanisms, which address the main challenges of the IoT: constrained and heterogeneous devices.

Acknowledgement.

This work has been partially supported by the SECURED project - which is co-funded by the European Commission under the ICT theme of FP7 (grant agreement no. 611458), and by the SHIELD project – which is co-funded by the European Commission under the call "Digital security: cybersecurity, privacy and trust" of the H2020 program (grant agreement no. 700199).

References.

1. Montero D, Yannuzzi M, Shaw A, Jacquin L, Pastor A, Serral-Gracia R, Lioy A, Risso F, Basile C, Sassu R, Nemirovsky M, Ciaccia F, Georgiades M, Charalambides S, Kuusijarvi J, Bosco F (2015) Virtualized security at the network edge: a user-centric approach. IEEE Communications Magazine 53:176-186. doi: 10.1109/mcom.2015.7081092
2. ETSI-NFV architectural framework (2013). http://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.01.01_60/gs_nfv002v010101p.pdf. Accessed 6 Sep 2016
3. Coker G, Guttman J, Loscocco P, Herzog A, Millen J, O'Hanlon B, Ramsdell J, Segall A, Sheehy J, Sniffen B (2011) Principles of remote attestation. International Journal of Information Security 10:63-81. doi: 10.1007/s10207-011-0124-7.
4. Jacquin L, Shaw A, Dalton C (2015) Towards trusted software-defined networks using a hardware-based Integrity Measurement Architecture. Netsoft

5. Pfaff B, Pettit J, Koponen T, Jackson E, Zhou A, Rajahalme J, Gross J, Wang A, Stringer J, Shelar P, Amidon K (2015) The design and implementation of open vswitch. NSDI'15 Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation:117-130.
6. McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J (2008). OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 38(2):69-74.
7. Basile C, Lioy A, Pitscheider C, Zhao S (2015) A formal model of policy reconciliation. Parallel Distributed and Network-Based Processing (PDP) 2015 23rd Euromicro International Conference:587-594. doi: 10.1109/PDP.20201515.42.
8. Kreps J, Narkhede N, Rao J (2011). Kafka: A distributed messaging system for log processing. Proceedings of the NetDB:1-7.
9. Apache Storm. http://storm.apache.org. Accessed 6 Sep 2016.
10. Bonafiglia R, Cerrato I, Ciaccia F, Nemirovsky M, Risso F (2015). Assessing the Performance of Virtualization Technologies for NFV: a Preliminary Benchmarking. 2015 Fourth European Workshop on Software Defined Networks:67-72. IEEE.
11. Montero D, Serral-Gracià R (2016). Offloading personal security applications to the Network Edge: A mobile user case scenario. Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International:96-101. IEEE.